Intelligent AI Applications and Autonomous Agents

Duration: 5 Days

Day 1: Foundations of Intelligent AI Applications

Module 1: AI Application Landscape and LLM Fundamentals

- Intelligent AI apps Foundation
- AI apps vs traditional software
- LLM capabilities and limits
- Tokens, context and memory
- Prompt engineering basics
- Choosing the right LLM

Lab Exercises:

1. Connect to an LLM API, send structured prompts and compare outputs across different prompt styles.
2. Build a simple Q&A app that accepts user input and returns LLM-generated responses.

Module 2: Building Intelligent Applications with LLMs

- Application architecture for AI apps
- Structured output generation
- Few-shot and chain-of-thought
- Handling hallucinations
- Conversation history management
- Streaming responses

Lab Exercises:

3. Build a multi-turn conversational app that maintains context across messages using conversation history.
4. Implement structured JSON output from an LLM and parse it into a usable application response.

Day 2: RAG and Knowledge-Augmented AI Applications

Module 3: Retrieval-Augmented Generation (RAG)

- Why RAG exists
- RAG architecture overview
- Document loaders and chunking
- Embedding models
- Vector databases
- Similarity search

Lab Exercises:

5. Build a RAG pipeline that loads documents, generates embeddings and stores them in a vector database.
6. Query the vector database and augment LLM prompts with retrieved chunks to generate grounded answers.

Module 4: Advanced RAG Techniques

- Hybrid retrieval
- Query rewriting
- Metadata filtering
- Re-ranking results
- RAG evaluation metrics
- Answer faithfulness

Lab Exercises:

7. Implement hybrid retrieval combining keyword and semantic search and compare accuracy with basic RAG.
8. Evaluate RAG pipeline quality using faithfulness and relevance metrics on a provided test dataset.

Day 3: Introduction to Autonomous AI Agents

Module 5: AI Agent Fundamentals

- What an AI agent is
- Agents vs chatbots vs apps
- ReAct framework
- Tool use and function calling
- Agent memory types
- Agent planning and reasoning

Lab Exercises:

9. Build a basic ReAct agent that reasons step by step and selects from a set of provided tools to answer a query.
10. Add tool calling to an LLM to enable it to perform web search and return grounded responses.

Module 6: Building Agents with LangChain and LangGraph

- LangChain agent architecture
- Tools and toolkits
- LangGraph state machines
- Nodes and edges

- Conditional routing
- Agent loop and termination

Lab Exercises:

11. Build a LangChain agent with three custom tools and observe how it selects and sequences tool calls.
12. Implement a LangGraph workflow with conditional edges that routes agent tasks based on classification output.

Day 4: Multi-Agent Systems and Agentic Workflows

Module 7: Multi-Agent Architecture

- Why multi-agent systems
- Orchestrator and worker agents
- Agent communication patterns
- Shared vs isolated memory
- CrewAI framework overview
- Role-based agent design

Lab Exercises:

13. Build a multi-agent system using CrewAI with an orchestrator agent delegating tasks to two specialist worker agents.
14. Implement shared memory between agents so context from one agent is available to the next in the pipeline.

Module 8: Agentic Workflow Design and Human-in-the-Loop

- Designing agentic workflows
- Long-running task management
- Human-in-the-loop patterns
- Interrupt and resume
- Agent error handling
- Logging and observability

Lab Exercises:

15. Build an agentic workflow with a human approval step that pauses execution and resumes after confirmation.
16. Add error handling and logging to an agent pipeline and test recovery behaviour when a tool call fails.

Day 5: Deployment, Safety and Real-World Application

Module 9: Deploying Intelligent AI Apps and Agents

- FastAPI for AI app deployment
- Containerising AI agents
- Scaling agent services
- Secrets and API key management
- Monitoring agent behaviour
- Cost control strategies

Lab Exercises:

17. Wrap an AI agent in a FastAPI endpoint and test it with real HTTP requests.
18. Add monitoring and usage logging to track token consumption and response times per agent call.

Module 10: Agent Safety, Guardrails

- Risks in autonomous agents
- Prompt injection and jailbreaking
- Output validation and guardrails
- Limiting agent permissions
- Responsible agentic AI
- Capstone project overview

Lab Exercises:

19. Implement input and output guardrails on an agent to block unsafe prompts and validate responses before delivery.
20. Capstone: build and present an end-to-end intelligent agent application combining RAG, tools and multi-agent coordination.